

**SISTEMA DE INFORMACIÓN PARA EL APOYO A LOS PROCESOS
ACADÉMICOS DE LOS ESTUDIANTES DEL CICLO NIVELATORIO DE LA
UNIVERSIDAD DEL MAGDALENA – SISNIV**

**LORENA PATRICIA LARA SÁNCHEZ
TATIANA RESTREPO BERMÚDEZ**

**UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SANTA MARTA D.T.C.H.
2006**

**SISTEMA DE INFORMACIÓN PARA EL APOYO A LOS PROCESOS
ACADÉMICOS DE LOS ESTUDIANTES DEL CICLO NIVELATORIO DE LA
UNIVERSIDAD DEL MAGDALENA – SISNIV**

**LORENA PATRICIA LARA SÁNCHEZ
TATIANA RESTREPO BERMÚDEZ**

**Proyecto de Memoria de Grado presentado como requisito parcial para optar
al título de Ingeniero de Sistemas**

**Director
Eduardo Ropaín Munive
Ingeniero de Sistemas**

**UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SANTA MARTA D.T.C.H.
2006**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Santa Marta, _____ de _____ de 2006

A Dios por ser la base y soporte de toda mi vida.

A mis padres Rosalinda y Elías por sus enseñanzas, brindarme siempre su gran amor, inmenso apoyo, comprensión, confiar tanto en mí y sobretodo, por darme tan excelente ejemplo, los amo y admiro demasiado. . .

A mis abuelos Helena y Rodrigo por ser como son y darnos lecciones de amor cada día.

A mis hermanos Rodrigo, Sebastián y Ronald por estar siempre ahí brindándome su apoyo, amor y hacerme sonreír todos los días.

A toda mi familia por siempre confiar en mí y brindarme su cariño y comprensión

Lorena.

Este trabajo se lo dedico a mis madres, Yolanda por haberme dado la vida, su coraje y fuerza. Todo es posible gracias a ti, a Mercedes por ser el árbol en donde siempre encuentro sombra y paz, gracias por quererme tanto, no hay recuerdo en el que no estés tú.

A mi abuelita, soñadora de mis éxitos, has llenado mi vida de ternura, nunca he de olvidar la dulzura de tus ojos.

A mi hermana por su protección y apoyo a lo largo del corto camino que hasta el momento ha sido mi vida.

A mi tía Miryam, juntas somos una familia un matriarcado, soy muy feliz al vivir con ustedes y gozar de su compañía.

Taliana

AGRADECIMIENTOS

Queremos agradecer a nuestros profesores, por su enseñanza y formación a lo largo de estos cinco años. En especial a nuestro director de tesis Eduardo Ropaín Munive por su gran dedicación, comprensión y amistad.

Al personal que labora en CIDS por colaborarnos en todo lo que necesitamos.

A nuestras amigas: Diana, Karem, Karina, Kelly, Yeilys y Ginna, por habernos acompañado en el proceso de crecer. Son demasiados los recuerdos, los errores, las alegrías y las tristezas que durante todo este tiempo hemos compartido juntas. Muchas gracias por darnos ánimos cuando lo necesitábamos.

Mil gracias a nuestros compañeros y amigos de Universidad: Luis Alfredo, gracias por tantos momentos, te queremos mucho; Ramiro, Jesús David, Osbert, Alex, Harold, Larry, Agustín, René y Enrique, por su apoyo, amistad y por tantas noches y amaneceres de estudio.

A Carlos Mario y Stanlee por compartir sus ideas, conocimientos y creatividad, significativos para la realización de este proyecto.

A nuestras familias por toda la comprensión y colaboración necesaria para conseguir la realización de este proyecto.

CONTENIDO

INTRODUCCIÓN	9
1. PRESENTACIÓN DEL PROYECTO	12
1.1. PLANTEAMIENTO DEL PROBLEMA	12

1.2. ANTECEDENTES	13
1.3. IMPORTANCIA Y JUSTIFICACIÓN	15
1.4. OBJETIVOS	16
1.4.1. Objetivo General	16
1.4.2. Objetivos Específicos	16
1.5. DELIMITACIÓN DEL ESPACIO TEMPORAL	17
2. MARCO REFERENCIAL	18
2.1. MARCO TEORICO	18
2.1.1. CICLO NIVELATORIO	18
2.1.2. BASES DE DATOS	18
2.1.3. INGENIERIA DE SOFTWARE	21
2.1.4. JAVA	26
2.1.5. JSP	28
2.1.6. UML	30
2.1.7. ARQUITECTURA CLIENTE/SERVIDOR	37
2.2. MARCO CONCEPTUAL	38
3. DISEÑO METODOLÓGICO	41
3.1. METODOLOGÍA	41
3.1.1. RECOLECCIÓN DE INFORMACIÓN	41
3.1.2. ANÁLISIS DE REQUERIMIENTOS	42
3.1.3. DISEÑO	51
3.1.4. CODIFICACIÓN	65
3.1.5. PRUEBA	65
4. CRONOGRAMA DE ACTIVIDADES	67
5. PRESUPUESTO	69
6. CONCLUSIONES	70

LISTA DE TABLAS

Tabla 1. Caso de uso: Configuración y Cambios	43
Tabla 2. Caso de uso: Administración de Docentes	44
Tabla 3. Caso de uso: Administración de Estudiantes	44
Tabla 4. Caso de uso: Registrar Calificaciones	45
Tabla 5. Caso de uso: Gestionar Inscripciones Universitarias	46

Tabla 6.	Caso de uso: Reportes e Informes	47
Tabla 7.	Entidad Estudiante	57
Tabla 8.	Entidad Docente.....	58
Tabla 9.	Entidad Asignatura.....	58
Tabla 10.	Entidad Depto	59
Tabla 11.	Entidad Ciudad.....	59
Tabla 12.	Entidad Grupo	59
Tabla 13.	Entidad Jornada	60
Tabla 14.	Entidad Estado.....	60
Tabla 15.	Entidad Tipo_pago	60
Tabla 16.	Entidad Fecha	61
Tabla 17.	Entidad Estudiante_grupo.....	61
Tabla 18.	Entidad Grupo_docente	61
Tabla 19.	Entidad Asign_Grupo_Dia_Hora.....	62
Tabla 20.	Entidad Nota	62
Tabla 21.	Entidad Matrícula	63
Tabla 22.	Entidad Habilitarpage.....	63
Tabla 23.	Entidad Usuario.....	63
Tabla 24.	Entidad Tipo_Usuario	64
Tabla 25.	Entidad Egresados.....	64
Tabla 26.	Cronograma de actividades	67
Tabla 27.	Presupuesto: Total	69
Tabla 28.	Presupuesto: Descripción de los gastos de personal.....	69
Tabla 29.	Presupuesto: Descripción de los equipos.....	69
Tabla 30.	Presupuesto: Descripción del Software	69
Tabla 31.	Presupuesto: Materiales de Papelería.....	69

LISTA DE FIGURAS

Figura 1.	Modelo de Ciclo de vida Lineal Secuencial	22
Figura 2.	Modelo de Ciclo de vida Incremental.....	23
Figura 3.	Modelo de Ciclo de Vida en Espiral.....	23
Figura 4.	Proceso de pruebas.....	25
Figura 5.	Caso de Uso: Principal	48

Figura 6. Caso de Uso: Configuración y Cambios.....	49
Figura 7. Caso de Uso: Administrar Docentes.....	49
Figura 8. Caso de Uso: Administrar Estudiantes	49
Figura 9. Caso de Uso: Gestionar Calificaciones (Docente)	50
Figura 10. Caso de Uso: Gestionar Calificaciones (SICIN)	50
Figura 11. Caso de Uso: Gestionar Inscripciones Universitarias	50
Figura 12. Caso de Uso: Generar Reportes	51
Figura 13. Arquitectura web de SISNIV	52
Figura 14. Interfaz de usuario. Página de inicio.	53
Figura 15. Interfaz de usuario. Mensajes de alerta, error y operación exitosa.....	53
Figura 16. Interfaz de usuario. Inserción de datos	54
Figura 17. Interfaz de usuario. Reportes.	55

LISTA DE ANEXOS

ANEXO A. Formato de Entrevistas

ANEXO B. Acuerdos Académicos

INTRODUCCIÓN

Con el propósito de contribuir con el mejoramiento de la calidad educativa en el Departamento y la región; La Universidad del Magdalena decide crear en el año 2001 y después de la reforma académica del mismo año, una alternativa para reforzar el nivel académico de los aspirantes con miras a alcanzar un mayor desempeño en el examen de admisión a la Universidad. Es entonces cuando surge el Ciclo Nivelatorio de La Universidad del Magdalena.

El objetivo principal de los Cursos Nivelatorios es llegar a aquellas personas que requieren mejorar su desempeño académico y reafirmar su vocacionalidad, proporcionándoles sólidas bases conceptuales que contribuyan a su crecimiento personal y por ende al desarrollo local y regional.

Para la Facultad de Estudios Generales, quién está a cargo de estos cursos de formación básica, se hace necesario manejar de forma adecuada y precisa toda la información referente a los mismos, tal como datos personales de los estudiantes, desempeño académico de éstos, información de grupos, horarios, docentes, etc. Por esta razón, por la importancia que han adquirido, gracias a los logros alcanzados por los cursos nivelatorios, surge entonces la necesidad de crear un sistema de información, que permita gestionar adecuadamente la información que se maneja en los mismos, contribuyendo con esto a agilizar los

procesos que dentro de ellos se realizan, mejorando su productividad y prestación de sus servicios.

1. PRESENTACIÓN DEL PROYECTO

Respondiendo a las necesidades de sistematización de los procesos en la Facultad de Estudios Generales, se desarrolló el Sistema de Información para el apoyo a los procesos académicos de los estudiantes del Ciclo Nivelatorio, con el fin de agilizar las tareas concernientes al mismo.

En razón a ello, se desarrollaron los diferentes módulos dentro del sistema, que permiten manejar la inscripción, matrícula y seguimiento académico de los estudiantes del Ciclo, así como la administración de información personal y académica de los docentes, de gran importancia para la coordinación de los cursos.

Con el fin de tener una visión más amplia acerca del sistema, el diseño del mismo se realizó implementando la metodología de modelado UML, y tratando hacer su manejo agradable a los usuarios, SISNIV se desarrolló utilizando el lenguaje de programación JSP, permitiendo crear un ambiente Web. El motor de base de datos utilizado fue ORACLE.

1.1. PLANTEAMIENTO DEL PROBLEMA

La coordinación del Ciclo Nivelatorio de la Universidad del Magdalena no cuenta con las herramientas necesarias para manipular adecuadamente la información y manejar de manera óptima cada uno de sus procesos, trayendo consigo el

excesivo y tedioso trabajo de forma manual por parte de los funcionarios encargados del Ciclo.

Se hace necesario entonces, implementar un sistema de información que permita solucionar inconvenientes tales como, la pérdida y redundancia de datos y documentos, lentitud e ineficiencia en la realización de los procesos y búsqueda de la información.

Con el desarrollo del Sistema de Información para el apoyo de los Procesos Académicos de los Estudiantes del Ciclo Nivelatorio se logrará resolver y se conseguirá entonces brindar acceso rápido a los datos, facilitando la búsqueda de información referente a estudiantes, jornadas y sus grupos, docentes adscritos al curso, etc., contando asimismo con las herramientas necesarias para que el registro de las calificaciones por parte de los docentes sea más ágil y sencillo. Además permitirá a la Facultad de Estudios Generales administrar procesos tales como asignación de horarios, grupos y docentes; de tal manera que se pueda garantizar la seguridad de los datos en todos los procesos, favoreciendo el buen desempeño de las labores y el mejoramiento de la productividad en la Facultad.

1.2. ANTECEDENTES

Producto de una profunda crisis financiera, administrativa y académica que mantuvo a la Universidad del Magdalena al borde del cierre definitivo en el año 1997; se inicia en la Institución un proceso de transformación en todos los órdenes, dando como resultado la refundación de la Universidad. Dentro del marco de este proyecto se da origen en el año 2001 al proceso de modernización académica integral, que con la creación de la Facultad de Estudios Generales

pretende lograr la excelencia en todos los procesos que en el Alma Máter se adelanten en aras de mejorar la calidad de la educación media, la cual presenta bajos índices de calidad en la región Caribe cuando se compara su nivel con los estándares nacionales e internacionales.

Perteneciente a la Facultad de Estudios Generales se encuentra el Ciclo Nivelatorio, con el cual se pretende lograr que los estudiantes mejoren su desempeño en el examen de admisión a la Universidad, además de disminuir los índices de deserción y mortalidad académica de los estudiantes en los primeros semestres, brindándoles las bases necesarias para iniciar su proceso de educación superior dentro del Alma Máter.

Al iniciar el desarrollo de los cursos del Ciclo Nivelatorio, éstos eran manejados por el Dependencia de Admisiones, Registro y Control Académico la cual se encargaba del proceso de matrículas de los estudiantes del Ciclo, sin embargo, dado el volumen de información manejada por esta dependencia, le fue encomendada esta tarea a la Facultad de Estudios Generales, ahora encargada de manejar en su totalidad la información referente al Ciclo Nivelatorio.

En un comienzo, la información fue manejada de forma manual, siendo almacenada en libros, y carpetas en archivos físicos, trayendo consigo muchas dificultades al tratar de acceder a los datos y el manejo adecuado de éstos. Actualmente se cuenta con la herramienta Excel, por medio de la cual se manejan los listados de grupos y estudiantes pertenecientes a ellos, pero se carece de una herramienta que permita a los docentes registrar el desempeño académico de los estudiantes de manera ágil, rápida y segura, así como la organización adecuada de los demás procesos relativos al Ciclo Nivelatorio.

1.3. IMPORTANCIA Y JUSTIFICACIÓN

Debido a los resultados obtenidos por los estudiantes que han cursado el Ciclo Nivelatorio y a la exaltación que ha tenido dicho curso de parte de evaluadores externos a la universidad, el número de aspirantes a cursarlos ha ido en aumento durante los últimos semestres, trayendo como consecuencia el crecimiento del volumen de información a manejar por La Facultad de Estudios Generales.

Por tal motivo, y dada la importancia que adquiere la información como herramienta para la planeación y toma de decisiones en toda organización, se hace necesario en la coordinación del Ciclo Nivelatorio, contar con un sistema de información que apoye los procesos que a partir de él se generan.

En la actualidad, los procesos que se llevan a cabo dentro del curso no se encuentran automatizados de manera adecuada, trayendo consigo inconvenientes en el manejo y acceso a la información para el personal administrativo del Ciclo Nivelatorio, dando como resultado la necesidad de crear un sistema de información que permita el registro y manejo de la información referente a la inscripción, matrícula, seguimiento y control académico de los estudiantes.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un Sistema de Información para el Ciclo Nivelatorio de la Universidad del Magdalena, que agilice los procesos que en él se realizan, tales como inscripción, seguimiento y control de sus estudiantes. Así como creación de grupos y asignación de docentes a los mismos.

1.4.2. Objetivos Específicos

- Determinar los requerimientos técnicos del sistema de información a través de la recolección y análisis de la misma, con el fin de establecer restricciones de diseño, criterios de validación, funciones o rendimiento del sistema, así como las interfaces que han de establecerse.
- Establecer los modelos lógicos y funcionales para construir el sistema, a través del diseño de entradas y salidas, interfaces, unido a las restricciones y procedimientos de control, utilizando el Lenguaje Unificado de Modelado UML.
- Diseñar y construir una base de datos robusta y segura que asegure un óptimo almacenamiento de la información de manera confiable, utilizando para ello el sistema manejador de Bases de Datos ORACLE.

- Desarrollar los módulos funcionales del software, tales como: Estudiantes, Docentes, Inscripciones Universitarias, Reportes, Grupos, utilizando el lenguaje de programación Jsp.
- Estudiar el comportamiento del software para asegurar la calidad de la información y su correcto funcionamiento.

1.5. DELIMITACIÓN DEL ESPACIO TEMPORAL

El Sistema de Información para el apoyo de los procesos académicos de los estudiantes del Ciclo Nivelatorio, será implementado en la Facultad de Estudios Generales de la Universidad del Magdalena, ubicada dentro de las instalaciones de la misma en la ciudad de Santa Marta. Se contará con SISNIV en cada periodo académico en el cual se programen cursos nivelatorios, además de que se le hagan actualizaciones para los cambios o creación de las actividades que se realicen en la Coordinación del Ciclo Nivelatorio.

2. MARCO REFERENCIAL

2.1. MARCO TEORICO

2.1.1. CICLO NIVELATORIO

El Ciclo Nivelatorio se constituye en un proceso académico propuesto para dar respuesta significativa y pertinente a la necesidad de mejorar la calidad y preparación académica de los jóvenes y adultos que no han alcanzado desarrollar las competencias requeridas para acceder a la Universidad del Magdalena. En este sentido pretende contribuir al mejoramiento de los procesos de admisiones y nivelación de los no admitidos en los programas académicos que ofrece la institución.

El Ciclo Nivelatorio es una propuesta de preparación académica orientada a los bachilleres que no accedieron a los programas profesionales de la Universidad del Magdalena, por obtener puntajes bajos en la prueba de ingreso, susceptibles de ser mejorados a través de un programa formativo con duración de un semestre, que propiciará el desarrollo de destrezas y habilidades.

2.1.2. BASES DE DATOS

- **Base de datos**

Es una colección de información, que está organizada de forma tal que su contenido sea fácilmente accesado, administrado y actualizado. Es decir, cualquier conjunto de datos organizados para su almacenamiento en la memoria de un computador, diseñado para facilitar su almacenamiento y acceso de una

forma estándar. Los datos suelen aparecer en forma de texto, números o gráficos. Los tipos más comunes de Base de Datos son las relacionales donde la información está definida de una forma en que pueda reorganizarse y accesarse de múltiples formas distintas; las distribuidas, con acceso desde diferentes puntos de una red y las orientadas a objetos, que clasifican la información sobre la base de clases y subclases.

Dentro de los elementos que conforman una base de datos se encuentran los archivos, el lenguaje de consultas, el diccionario de datos y el sistema de gestión de la base de datos.

- **Sistema de Gestión de Base de Datos**

Un sistema de gestión de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar esos archivos, con el fin de simplificar los esfuerzos de desarrollo de software al proporcionar servicios genéricos adaptables a aplicaciones variables; además, un sistema de bases de datos contiene un método general para mantener relaciones entre los elementos de datos.¹

Un objetivo importante de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo, para que el sistema sea manejable, los datos se deben extraer eficientemente. Este requerimiento ha llevado al diseño de estructuras de datos complejas para la representación de datos en la base de datos. Puesto que muchos usuarios de sistemas de bases de datos no tienen experiencia en computadores, se les

¹ KORTH, Henry F. Fundamentos de Bases de Datos. 2da. Edición. Madrid: Mc. Graw Hill, 1994, p. 4

esconde la complejidad a través de diversos niveles de abstracción para simplificar su interacción con el sistema.²

- **Base de datos Relacional**

Las bases de datos se pueden clasificar de acuerdo a su modelo de administración de datos. A raíz de esto, han surgido cuatro modelos de bases de datos: modelo en Red, modelo Jerárquico, modelo Orientado a Objetos y modelo Relacional, siendo este último el más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Una base de datos relacional es un conjunto finito de relaciones, donde cada una de ellas contiene un esquema relacional y un cuerpo correspondiente. Los nombres de las relaciones deben ser distintivos y dentro de cada relación, los nombres de los atributos también deben ser distintivos.³

El modelo relacional utiliza una estructura particularmente sencilla cuando se compara con sus predecesores de red y jerárquicos. En el modelo relacional, una base de datos es un conjunto de tablas, cada una de ellas estructuralmente semejante a un archivo de registros de longitud fija. Debido a este conocido formato, el modelo relacional es accesible a una amplia variedad de usuarios no técnicos.

El modelo relacional también disfruta de la distinción de una rigurosa base matemática para sus operaciones. Comenzando con una descripción de teoría de conjunto de los elementos básicos, muchas características del modelo relacional se siguen como deducciones matemáticas. Por el contrario, los modelos

² KORTH, Henry F. Fundamentos de Bases de Datos. 2da. Edición. Madrid: Mc. Graw Hill, 1994, p. 5

³ JOHNSON, James. Bases de datos. Modelos, lenguajes, diseño. México: Editorial Oxford, 2000, p 31.

anteriores proceden con descripciones de comportamiento de las operaciones de las bases de datos.

2.1.3. INGENIERIA DE SOFTWARE ⁴

La ingeniería de software es la disciplina que se ocupa de la aplicación de la teoría, conocimiento y práctica para la construcción eficaz y eficiente de sistemas de software que satisfagan requerimientos de usuarios y clientes. Es aplicable a sistemas de escala pequeña, media y grande. El ciclo de vida del software incluye análisis de requerimientos, especificación, diseño, construcción, pruebas, operación y mantenimiento del software.

La ingeniería de software emplea métodos, procesos, técnicas y mediciones. Se beneficia del uso de herramientas para el desarrollo de software, para el análisis y modelaje, para el aseguramiento y control de la calidad, para la administración de la evolución y para la reutilización del software. El desarrollo de software requiere habilidades individuales y de grupo, uso de metodologías y seguimiento de procesos de producción.

- **Modelos de Ciclo de Vida de proyectos de software**

El software al igual que todos los sistemas complejos, evoluciona con el tiempo. Los requisitos de gestión y de productos a menudo cambian conforme a que el desarrollo proceda haciendo que el camino que lleva al producto final no sea real; las estrictas fechas tope del mercado hacen que no sea posible finalizar un producto completo, por lo que se debe introducir una versión limitada para cumplir la presión competitiva y de gestión; se comprende perfectamente el conjunto de

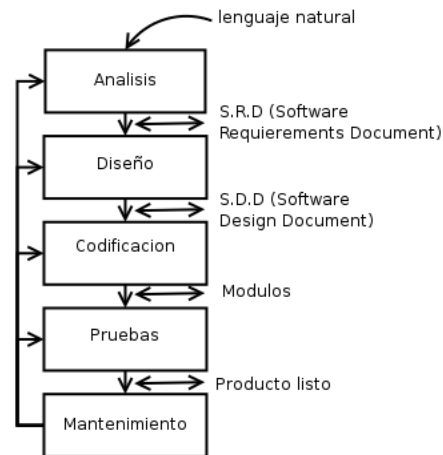
⁴ SOMMERVILLE, Ian. Ingeniería de Software. 6ª. Edición. Pearson Educación. p 288

requisitos de productos centrales o del sistema, pero todavía se tienen que definir los detalles de extensiones del producto o sistema. En estas y en otras situaciones similares, los ingenieros del software necesitan un modelo de proceso que se haya diseñado explícitamente para acomodarse a un producto que evolucione con el tiempo.

- Modelo Lineal Secuencial

El ciclo de vida clásico, básico o modelo de cascada considera las fases del desarrollo de software una después de la otra, con un enfoque sistemático, donde cada fase empieza donde termina la anterior.

Figura 1. Modelo de Ciclo de vida Lineal Secuencial

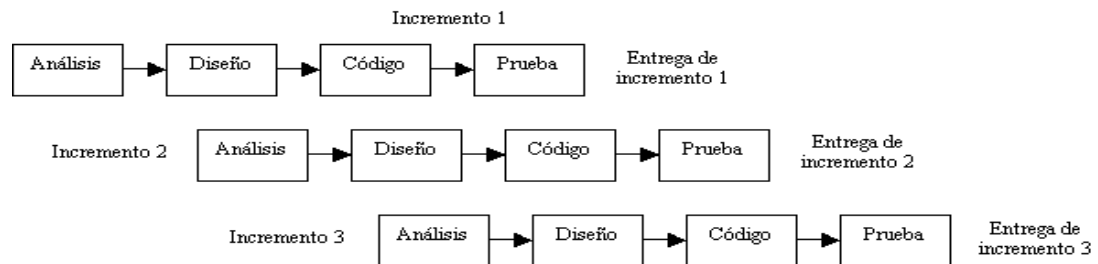


- Modelo Incremental

Combina elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. Es un modelo donde se realiza análisis, diseño, se construye una primera versión, con lo esencial del sistema, se prueba y se entrega, luego se repite se analiza, se diseña las mismas funciones más completas, se construye y se entrega una nueva versión, y se repite hasta tener

un sistema lo más completo posible. Cada incremento debe entregar un producto operacional y es revisado por el usuario para aportar más sobre las nuevas versiones.

Figura 2. Modelo de Ciclo de vida Incremental

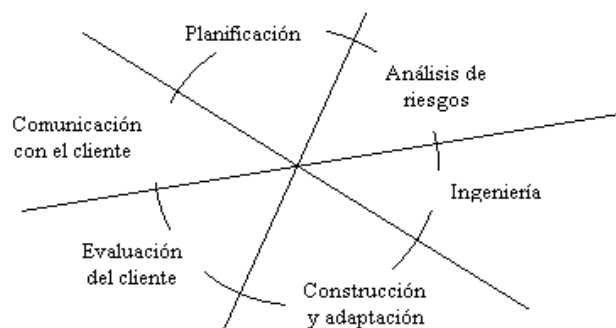


- Modelo Espiral

Es un modelo que acompaña la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales.

Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas de la ingeniería del sistema.

Figura 3. Modelo de Ciclo de Vida en Espiral



- **Especificación Del Software**

- Recolección y Análisis De Requerimientos

En esta actividad, el personal del desarrollo técnico del software trabaja con los clientes y los usuarios finales del sistema para determinar el dominio de la aplicación, cuáles servicios debe proveer el sistema, el desempeño requerido del sistema, las restricciones de hardware, etc.

Este análisis puede dividirse en las siguientes cinco actividades de esfuerzo:

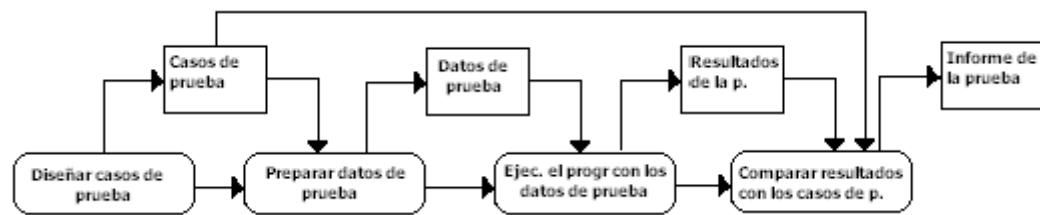
- Reconocimiento del problema. Reconocer los elementos básicos del problema, tal y como los percibe el usuario o el cliente.
- Evaluación y síntesis. Definir todos los objetos de datos observables externamente, evaluar el flujo y contenido de la información y elaborar todas las funciones del software. Esta actividad continúa hasta que el analista y el cliente se sienten seguros de que se puede especificar adecuadamente el software para las fases posteriores de desarrollo.
- Modelado. Durante la actividad de evaluación y síntesis de la solución, el analista crea modelos del sistema en un esfuerzo de entender mejor el flujo de datos y de control, el tratamiento funcional, el comportamiento operativo y el contenido de la información.
- Especificación. El modelo sirve como fundamento para el diseño del software y como base para la creación de una especificación del software.
- Revisión. Consiste en llevar adelante la actividad de revisar las especificaciones del software para considerar las mas adecuadas.

- Prueba de defectos

La meta de las pruebas de defectos es exponer los defectos latentes en un sistema de software antes de entregar el sistema. Esto contrasta con las pruebas de validación que pretenden demostrar que un sistema cumple su especificación.

Las pruebas de validación requieren que el sistema se desempeñe correctamente utilizando casos de pruebas dados. Una prueba de defectos exitosa es una prueba que provoca que el sistema se desempeñe incorrectamente y así exponer un defecto. Esto enfatiza un hecho importante acerca de las pruebas. Demuestra la presencia, no la ausencia, de fallas en el programa.

Figura 4. Proceso de pruebas



• Administración de Proyectos de Software

La Gestión de Proyectos tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un Sistema de Información. Como consecuencia de este control es posible conocer en todo momento qué problemas se producen y resolverlos o paliarlos de manera inmediata.

Un proyecto grande puede ocupar varios equipos de programadores, en cuyo caso cada uno tiene la responsabilidad de su estructura interna y de sus relaciones con los demás. Independientemente de la estructura del equipo, deben limitarse a no más de cinco a siete miembros. Esta restricción controla el número de trayectorias de comunicación en el equipo y permite una coordinación eficiente de las actividades.

La gestión de un proyecto es una tarea vital para el éxito del mismo. Dentro de la gestión, la planificación juega un importante rol, debido a que es en esta etapa

donde se debe hacer las asignaciones de los recursos disponibles, para lo que es necesario estimar costos y plazos para el proyecto a realizar.

La planificación de un proyecto se basa en una buena estimación del esfuerzo requerido para realizarlo, y para apoyar esta difícil tarea, se han desarrollado varios métodos que han encontrado aceptación comercial en forma creciente en la planificación del desarrollo de software.

2.1.4. JAVA

En 1991, la empresa **Sun Microsystems** crea el lenguaje **Oak** (de la mano del llamado proyecto **Green**), mediante el cual se pretendía crear un sistema de televisión interactiva. Este lenguaje sólo se llegó a utilizar de forma interna. Su propósito era crear un lenguaje independiente de la plataforma y para uso en dispositivos electrónicos.

Se intentaba con este lenguaje paliar el problema fundamental del C++; que consiste en que al compilar se produce un fichero ejecutable cuyo código sólo vale para la plataforma en la que se realizó la compilación. Sun deseaba un lenguaje para programar pequeños dispositivos electrónicos. La dificultad de estos dispositivos es que cambian continuamente y para que un programa funcione en el siguiente dispositivo aparecido, hay que rescribir el código. Por eso Sun quería crear un lenguaje independiente del dispositivo.

En 1995 pasa a llamarse **Java** y se da a conocer al público. Adquiere notoriedad rápidamente. Java pasa a ser un lenguaje totalmente independiente de la plataforma y a la vez potente y orientado a objetos. Esa filosofía y su facilidad para

crear aplicaciones para redes TCP/IP ha hecho que sea uno de los lenguajes más utilizados en la actualidad.

La versión actual de Java es el llamado Java 2. Sus ventajas sobre C++ son:

- Su sintaxis es similar a C y C++
- No hay punteros (lo que le hace más seguro)
- Totalmente orientado a objetos
- Muy preparado para aplicaciones TCP/IP
- Implementa excepciones de forma nativa
- Es interpretado (lo que acelera su ejecución remota, aunque provoca que las aplicaciones Java se ejecuten más lentamente que las C++ en un ordenador local).
- Permite multihilos
- Admite firmas digitales
- Tipos de datos y control de sintaxis más rigurosa
- Es independiente de la plataforma

La última ventaja (quizá la más importante) se consigue ya que el código Java no se compila, sino que se **precompila**, de tal forma que se crea un código intermedio que no es ejecutable. Para ejecutarle hace falta pasarle por un intérprete que va ejecutando cada línea. Ese intérprete suele ser la máquina virtual de Java.

- **Tipo de Aplicaciones Java**

- Applet: Son programas Java pensados para ser colocados dentro de una página Web. Pueden ser interpretados por cualquier navegador con capacidades Java.

Estos programas se insertan en las páginas usando una etiqueta especial (como también se insertan vídeos, animaciones flash u otros objetos). Los applets son programas independientes, pero al estar incluidos dentro de una página Web las reglas de éstas le afectan. Normalmente un applet sólo puede actuar sobre el navegador.

- Aplicaciones de Consola. Son programas independientes al igual que los creados con los lenguajes tradicionales.
- Aplicaciones Gráficas. Aquellas que utilizan las clases con capacidades gráficas (como awt por ejemplo).
- Servlets. Son aplicaciones que se ejecutan en un servidor de aplicaciones Web y que como resultado de su ejecución resulta una página Web.⁵

2.1.5. JSP

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas Web con programación en Java.

Se trata de una idea que puede convivir perfectamente con los servlets (de hecho refuerza esa tecnología), pero cuyo método de trabajo es distinto. Una página JSP es una página Web normal (sólo que con extensión .jsp) a la que se le puede añadir código java utilizando unas etiquetas especiales dentro del código de la página. Estas etiquetas son traducidas por el servidor de aplicaciones al igual que traduce el código de un servlet. Las etiquetas JSP comienzan por `<%` y terminan por `%>`

⁵ SANCHEZ, Jorge. Java 2. Castilla y León, España. 2004. En: www.jorgesanchez.net

Es una técnica cada vez más popular ya que posee la potencia de Java (al mismo nivel que los servlets), pero con la gracia de escribir directamente el código en una página web, lo que facilita su diseño. Su compatibilidad es mucho mayor y su creación es más simple. Realmente en la práctica todo JSP se convierte en un servlet, por lo que se necesita el mismo sistema de archivos (carpeta WEB-INF, web.xml, classes, lib, etc.). Por lo que las páginas JSP se almacenan en la raíz de la aplicación y sus librerías comprimidas como jar en la carpeta lib (todas las librerías deben almacenarse ahí, las del kit de Java no porque el servidor de aplicaciones ya las incorporará).

- **Ciclo de Vida de una Página JSP**

Toda página JSP atraviesa una serie de etapas:

- Comienza con el código nativo HTML-JSP. Es el código escrito por el programador, mezcla de HTML clásico e instrucciones Java.
- La página JSP se transforma en el Servlet equivalente. Esto ocurre tras la petición de la página por parte del cliente. Se genera pues un archivo class dispuesto a ser ejecutado cuando haga falta, el servidor lo almacena para su posterior uso, por lo que en peticiones siguientes ya no se compilará: a no ser que se detecten cambios en el código original
- Se carga la clase para cada petición entrante, el ejemplar, que gestiona la petición http del cliente concreto.
- Finalmente la ejecución del servlet da lugar al código HTML que es el que recibe el cliente.⁶

⁶ SANCHEZ, Jorge. Java 2. Castilla y León, España. 2004. En: www.jorgesanchez.net

2.1.6. UML

El lenguaje Unificado de Modelado (Unified Modeling Lenguaje, UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

UML es apropiado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la Web, e incluso para sistemas empotrados de tiempo real muy exigentes. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas. Aunque sea expresivo, UML no es difícil de aprender ni de utilizar.

UML es sólo un lenguaje y por tanto es tan sólo una parte de un método de desarrollo de software. UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

El vocabulario de construcción de UML incluye tres clases de bloques de construcción: Elementos, Relaciones y Diagramas. Los elementos son abstracciones que son ciudadanos de primera clase en un modelo; las relaciones ligan estos elementos entre sí; los diagramas agrupan colecciones interesantes de elementos. De este último bloque de construcción, el lenguaje de modelamiento estandariza 9 tipos de diagramas para representar gráficamente un sistema desde distintos puntos de vista.⁷

⁷ JACOBSON, Ivar. El Lenguaje Unificado de Modelamiento. Addison Wesley. Madrid 1999, p 11

- **Diagramas UML**

Un diagrama es la representación gráfica de un conjunto de elementos, visualizado la mayoría de las veces como un grafo conexo de elementos y relaciones. Los diagramas se dibujan para visualizar un sistema desde diferentes perspectivas. Un diagrama representa una vista resumida de los elementos que constituyen un sistema.

- Diagrama de casos de uso

Un diagrama de casos de uso es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo las relaciones entre ellos, representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Todo sistema tiene como mínimo un diagrama (caso de uso principal), que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Estos diagramas facilitan que los sistemas, subsistemas y clases sean abordables y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado. Los diagramas de casos de uso también son importantes, para probar sistemas ejecutables a través de ingeniería directa y para comprender sistemas ejecutables a través de ingeniería directa y para comprender sistemas ejecutables a través de ingeniería inversa.⁸

⁸ JACOBSON, Ivar. El Lenguaje Unificado de Modelamiento. Addison Wesley. Madrid 1999, p 202

Los elementos de los diagramas de Caso de Uso son:

Sistema: Un sistema en un diagrama de caso de uso es descrito como una caja; el nombre del sistema aparece arriba o dentro de la caja. Ésta también contiene los símbolos para los casos de uso del sistema.

Actores: Un actor es alguien o algo que interactúa con el sistema; es quien utiliza el sistema. Por la frase "interactúa con el sistema" se debe entender que el actor envía a o recibe del sistema unos mensajes o intercambia información con el sistema. En pocas palabras, el actor lleva a cabo los casos de uso. Un actor puede ser una persona u otro sistema que se comunica con el sistema a modelar. Gráficamente se representa con la figura de "stickman".

Casos de uso: Un caso de uso representa la funcionalidad completa tal y como la percibe un actor. Un caso de uso en UML es definido como un conjunto de secuencias de acciones que un sistema ejecuta y que permite un resultado observable de valores para un actor en particular. Gráficamente se representan con una elipse y tiene las siguientes características:

- Un caso de uso siempre es iniciado por un actor.
- Un caso de uso provee valores a un actor.
- Un caso de uso es completo.

Relaciones: Entre los elementos de un diagrama de casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas o no entre ellos:

- Asociación: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

- Dependencia: Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.
- Generalización: Este tipo de relación es uno de los más utilizados y está orientado exclusivamente para casos de uso (y no para actores). Cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>). Se recomienda usar *extends* cuando un caso de uso es similar a otro; y utilizar *uses* cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

○ Diagrama de Clases

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los diagramas más comunes en el modelado de sistemas orientados a objetos. Los diagramas de clases cubren la vista de diseño estática de un sistema.

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.⁹

Los diagramas de clases contienen normalmente los siguientes elementos:

⁹ JACOBSON, Ivar. El Lenguaje Unificado de Modelamiento. Addison Wesley. Madrid 1999, p 93

Clases: Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio. En UML, una clase es representada por un rectángulo que posee tres divisiones

Interfaces: Una interfaz es una colección de operaciones que especifican un servicio de una clase o componente. Por lo tanto, una interfaz describe el comportamiento visible externamente de ese elemento.

Colaboraciones: Una colaboración define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Una clase puede participar en varias colaboraciones.

Relaciones: En los diagramas de clases se emplean las mismas relaciones antes mencionadas en los diagramas de casos de uso.

- Diagrama de secuencia

Un diagrama de secuencia es un dibujo creado de manera rápida y fácil, que muestra los eventos de entrada y salida relacionados con el sistema estudiado. Muestra para cada escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre sistemas.

El Diagrama de Secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte

superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.¹⁰

- Diagrama de objetos

Un diagrama de objetos muestra un conjunto de objetos y sus relaciones. Los diagramas de objetos representan instantáneas de instancias de los elementos encontrados en los diagramas de clases. Estos diagramas cubren la vista de diseño estática o la vista de procesos estática de un sistema como lo hacen los diagramas de clases, pero desde la perspectiva de casos reales o prototípicos.

- Diagrama de colaboración

Un diagrama de colaboración es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. El Diagrama de Colaboración presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Mientras que el diagrama de secuencia se centra en la secuencia cronológica del escenario que estamos modelando, el diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario. Los objetos se conectan por medio de enlaces, cada enlace representa una instancia de una asociación entre las clases implicadas. El enlace muestra los mensajes enviados entre los objetos, el tipo de mensaje, y la visibilidad de un objeto con respecto a los otros.

- Diagrama de estados

¹⁰ Popkin Software and Systems. Modelado de Sistemas con UML.
En: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>

El diagrama de estados muestra una máquina de estados, que consta de estados, transiciones, eventos y actividades. Los diagramas de estado cubren la vista dinámica de un sistema.

Un diagrama de Estados muestra el ciclo de vida de un objeto: qué eventos experimenta, sus transiciones a los estados en los que se encuentra entre estos eventos. No es necesario ilustrar todos los posibles eventos; si surge un evento que no está representado en el diagrama, se ignora el evento por lo que el diagrama de estados se refiere. Por lo tanto, se puede crear un diagrama de estado que describa el ciclo de vida de un objeto a un nivel de detalle arbitrariamente simple o complejo, dependiendo de nuestras necesidades.¹¹

Son especialmente importantes en el modelado del comportamiento de una interfaz, una clase o una colaboración y resaltan el comportamiento dirigido por eventos de un objeto, lo cual es especialmente útil en el modelado de sistemas reactivos.

- Diagrama de actividades

Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema. Los diagramas de actividades cubren la vista dinámica de un sistema. Son especialmente importantes al modelar el funcionamiento de un sistema y resaltan el flujo de control entre objetos.

- Diagrama de componentes

¹¹ LARMAN, Craig. UML Y PATRONES. Una introducción al análisis y diseño orientado a objetos y el proceso unificado. 2da. Edición. Prentice Hall. Madrid 2003, p 114

El diagrama de componentes muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones.

- Diagrama de despliegue

Muestra la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Los diagramas de despliegue cubren la vista de despliegue estática de una arquitectura. Se relacionan con los diagramas de componentes en que un nodo incluye, por lo común, uno o más componentes.¹²

2.1.7. ARQUITECTURA MULTICAPAS

La mayoría de las aplicaciones en la Internet operan con una arquitectura de tres capas: capa de presentación, de aplicación y de datos. En la capa de presentación se encuentran las interfaces de la aplicación para el usuario, en el caso de una aplicación Web, esta capa la compone el browser que permite visualizar las páginas Web enviadas por el servidor.

La capa de presentación se encarga de obtener los datos que el usuario ingresa, enviar estos a la capa de aplicación para su procesamiento, recibir los resultados del procesamiento de la aplicación que se hizo en la capa intermedia y presentar los resultados al usuario.

La capa de aplicación, es la capa intermedia que provee la infraestructura que enlaza las demás capas. Su función es recibir los datos del usuario enviados por la capa de presentación, procesarlos ejecutando la lógica del negocio de la

¹² JACOBSON, Ivar. El Lenguaje Unificado de Modelamiento. Addison Wesley. Madrid 1999, p 22

aplicación interactuando con la capa de datos y enviar el resultado del proceso a la capa de presentación.

La capa de los datos contiene, como su nombre lo indica, los depósitos de datos, sean estos sistemas de archivos o bases de datos y los mecanismos para su acceso.¹³

2.2. MARCO CONCEPTUAL

CICLO NIVELATORIO. Curso ofrecido por la Facultad de estudios Generales en el cual se capacita a los estudiantes que aspiran ingresar a la Universidad del Magdalena en las ciencias básicas, con el fin de obtener un mejor desempeño académico una vez ingresado a ésta.

CICLO DE VIDA DEL SOFTWARE. Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas: análisis, diseño, codificación, pruebas y mantenimiento.

BASE DE DATOS. Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

JAVA. Lenguaje desarrollado por Sun Microsystems para la elaboración de aplicaciones exportables a la red y capaces de operar sobre cualquier plataforma

¹³ ECUANET. Pioneros y Líderes en Internet. Ecuador. 2003
URL: <http://revista.ecua.net.ec>

a través, normalmente, de visualizadores WWW. El programa Java se descarga desde el servidor Web y lo interpreta un programa que se ejecuta en el equipo que contiene el explorador de Web.

JSP. (Java Server Page) Página de Servidor Java. Se refiere a un tipo especial de páginas HTML, en las cuales se insertan pequeños programas que corren sobre Internet, comúnmente denominados scripts, se procesan en línea para finalmente desplegar un resultado al usuario en forma de HTML. Por lo general dichos programas hacen consultas a bases de datos y dependiendo del resultado, se despliega la información que se muestra a cada usuario de manera individual.

MODELO: Un modelo es la representación abstracta de una especificación, un diseño o un sistema desde un punto de vista particular. A menudo se representa gráficamente mediante uno o más diagramas. Tiene como propósito expresar la esencia de algunos aspectos de lo que se está haciendo, sin especificar detalles innecesarios. Su propósito es permitir a los implicados en el desarrollo, pensar sobre los problemas y soluciones sin desviarse de los objetivos.

SISTEMA. Es un conjunto de elementos materiales, o inmateriales, (hombres, maquinas, métodos, reglas, etc.) relacionados que transforman, mediante procesos, elementos (entradas) en otros nuevos elementos (salidas).

SOFTWARE. Son todos los documentos asociados y la configuración de los datos que se necesitan para hacer que estos programas operen de manera correcta. Por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, u sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica como utilizar el sistema y, en cuanto a los productos de software, sitios Web que permitan a los usuarios descargar la información de productos recientes.

SISTEMA DE INFORMACIÓN: Un sistema de información se puede definir como el conjunto de funciones y procedimientos encaminados a la captación, desarrollo, recuperación, almacenamiento, etc., de información en el seno de una organización, para apoyar la toma de decisiones y el control de la misma.

UML: El Lenguaje estándar de Modelamiento Unificado (Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML proporciona una forma sencilla, mediante diagramas e iconografías, de escribir planos de software. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

3. DISEÑO METODOLÓGICO

3.1. METODOLOGÍA

Para el desarrollo del Sistema de Información SISNIV se empleó el modelo en cascada, el cual se caracteriza por un refinamiento del modelo por etapas, en donde se realizan ciclos de realimentación de las mismas, con el objetivo de minimizar el costo de trabajo del proyecto; comienza con la captura y análisis de requisitos, pasando posteriormente al diseño, codificación, pruebas, revisión y mantenimiento.

3.1.1. RECOLECCIÓN DE INFORMACIÓN

En esta primera fase fue recopilada la información necesaria para establecer los requerimientos del sistema, a través de entrevistas al personal administrativo y la revisión de documentos de la Facultad de Estudios Generales, con el fin de identificar los procesos relacionados con el Ciclo nivelatorio; así como los requerimientos funcionales y no funcionales con los que debe cumplir el sistema de información.

• Entrevistas

En la primera etapa de Recolección y Análisis de Requisitos se realizaron varias entrevistas con el fin de conocer detalladamente los procesos que se llevan a acabo dentro de la oficina de Estudios Generales de la Universidad del

Magdalena, en relación a los cursos nivelatorios que ésta ofrece. Las entrevistas fueron realizadas al personal encargado del manejo administrativo de los mismos.

En cada una de las entrevistas los funcionarios dieron a conocer las dificultades y necesidades encontradas en el manejo anteriormente dado a los procesos que éstos realizaban, debido a la carencia de un sistema de información que sirviera de apoyo y contribuyera al óptimo desarrollo de las actividades administrativas y académicas. Ver Anexo A.

- **Revisión de documentos**

Mediante la revisión de documentos como: acuerdos académicos, folletos de promoción, reportes generados de manera manual, formularios del curso nivelatorio, etc., proporcionados por la coordinadora del mismo, se obtuvo una visión más amplia y detallada de los requerimientos del software. Ver Anexo B.

3.1.2. ANÁLISIS DE REQUERIMIENTOS

Cumplido el proceso de recolección de información, se inicia el estudio de la información recopilada tanto técnica como general, necesaria para definir los requerimientos, funciones y restricciones del software. Así mismo, se definen las entradas y salidas del sistema, los actores que intervienen en éste y las interfaces entre los elementos del sistema y el dominio de la información. Dentro de los actores se pudieron identificar:

- Admin. Este usuario tiene amplio control sobre las configuraciones del sistema, así mismo podrá tener acceso a las funciones de otros usuarios, de manera que las pueda modificar y deshacer.

- SICIN. Es el operador o digitador del sistema. Posee ciertos privilegios sobre la información manejada a través del sistema, sin embargo no tiene acceso a operaciones que involucren cambios fundamentales sobre el sistema.
- Monitor. Este usuario es el consultor del sistema, sólo tiene acceso a las salidas de información y no puede realizar cambios sobre éstas.
- Docente. El usuario Docente sólo posee permiso de ingresar al modulo de Calificaciones, siempre y cuando haya grupos a su cargo.

- **Requerimientos específicos**

En este proceso se describieron los servicios que debía proveer el sistema a sus usuarios, de acuerdo con las necesidades anteriormente planteadas por ellos. Mediante la especificación de los requerimientos, se logró identificar el producto de software, así como los factores generales que afectan a éste y a los requerimientos.

Especificación de los casos de uso principales:

Tabla 1. Caso de uso: Configuración y Cambios

No.	CU – 01
Nombre Caso de Uso:	CONFIGURACIÓN Y CAMBIOS
Necesario/Deseable	Necesario
Actor Principal:	Admin.
Resumen:	Permite configurar, ingresar parámetros y realizar cambios de configuración.
Precondiciones:	El usuario se identifica y autentica.
Postcondiciones:	
Flujo Básico:	<ol style="list-style-type: none"> 1. Una vez el usuario ha ingresado a esta funcionalidad, el sistema presenta al usuario una serie de opciones: <ol style="list-style-type: none"> 1.1. Definir Fechas 1.2. Administración de Grupos. 1.3. Definir Costo Matrícula 2. El usuario selecciona una de las opciones. 3. El sistema remite al usuario a otra página, dependiendo de la opción escogida.
Flujos Alternativos:	

Tabla 2. Caso de uso: Administración de Docentes

No.	CU – 02
Nombre Caso de Uso:	ADMINISTRACIÓN DE DOCENTES
Necesario/Deseable	Necesario
Actor Principal:	ADMIN
Resumen:	Permite organizar y administrar los docentes con sus respectivos horarios y cursos a cargo.
Precondiciones:	El usuario se identifica y autentica.
Postcondiciones:	
Flujo Básico:	<ol style="list-style-type: none"> 1. Una vez el usuario ha ingresado a esta funcionalidad, el sistema muestra las opciones: <ol style="list-style-type: none"> 1.1. Ingresar Nuevo Docente 1.2. Modificar Datos Docente. 2. El usuario selecciona una de las opciones presentadas. 3. El sistema remite al usuario a otra página según la opción escogida.
Flujos Alternativos:	
Frecuencia:	Frecuentemente

Tabla 3. Caso de uso: Administración de Estudiantes

No.	CU - 03
Nombre Caso de Uso:	ADMINISTRACIÓN DE ESTUDIANTES
Necesario/Deseable	Necesario
Actor Principal:	SICIN
Resumen:	Permite al usuario inscribir, matricular y modificar datos de los estudiantes.
Precondiciones:	El usuario se identifica y autentica.
Postcondiciones:	
Flujo Básico:	<ol style="list-style-type: none"> 1. Una vez el usuario ha ingresado a esta funcionalidad, el sistema muestra las opciones: <ol style="list-style-type: none"> 1.1. Nueva Inscripción. 1.2. Modificar Datos. 1.3. Registrar Matrícula. 2. El usuario selecciona una de las opciones presentadas. 3. El sistema remite al usuario a otra página según la opción escogida.
Flujos Alternativos:	
Frecuencia:	Frecuentemente

Tabla 4. Caso de uso: Registrar Calificaciones

No.	CU – 04
Nombre Caso de Uso:	REGISTRAR CALIFICACIONES (DOCENTE)
Necesario/Deseable	Necesario
Actor Principal:	DOCENTE
Resumen:	Permite registrar y modificar las calificaciones de los estudiantes.
Precondiciones:	El docente se identifica y autentica.
Postcondiciones:	Se actualiza la Base de Datos.
Flujo Básico:	<ol style="list-style-type: none"> 1. Una vez el docente se ha registrado en la página inicial, le son mostrados los grupos asignados a él en el Ciclo Nivelatorio. El docente escoge el grupo a digitar notas. Se ejecuta el caso de uso consultar Estudiantes por grupo, y se muestran seis campos por estudiantes, tres de ellos habilitados para ingresar notas. 2. Se da la orden de Guardar 3. El sistema verifica que los datos ingresados se encuentren dentro de los parámetros establecidos por la Facultad de Estudios Generales. 4. Se almacenan las notas ingresadas.
Flujos Alternativos:	
Frecuencia:	

Tabla 5. Caso de uso: Gestionar Inscripciones Universitarias

No.	CU – 05
Nombre Caso de Uso:	GESTIONAR INSCRIPCIONES UNIVERSITARIAS
Necesario/Deseable	Necesario
Actor Principal:	SICIN
Resumen:	Permite registrar el programa académico en el cual los estudiantes del ciclo nivelatorio desean ingresar a la Universidad.
Precondiciones:	El usuario se identifica y autentica.
Postcondiciones:	Se actualiza la Base de Datos.
Flujo Básico:	<ol style="list-style-type: none"> 1. Una vez el usuario ha ingresado a esta funcionalidad, el sistema muestra las siguientes opciones: <ol style="list-style-type: none"> 1.1. Individual. 1.2. Grupales. 1.1. <u>Individual:</u> <ol style="list-style-type: none"> 1.1.1. El usuario ingresa el tipo y número del documento de identificación del estudiante y se ejecuta el Caso de Uso: Consultar Datos de Estudiante. 1.1.2. El usuario selecciona el programa académico al cual se inscribió el estudiante. 1.2. <u>Grupales:</u> <ol style="list-style-type: none"> 1.2.1. El usuario selecciona el grupo al cual se le registrarán las inscripciones universitarias. 1.2.2. Se ejecuta el caso de uso Consultar Estudiantes por grupo y se muestra un campo frente a cada estudiante, para seleccionar el programa académico escogido por éste. 2. Se da la opción de guardar
Flujos Alternativos:	<ol style="list-style-type: none"> 1.1.a. Si el Número de documento de identificación no existe en el sistema, se muestra un mensaje informando que el estudiante no pertenece al Ciclo Nivelatorio.
Frecuencia:	

Tabla 6. Caso de uso: Reportes e Informes

No.	CU - 06
Nombre Caso de Uso:	REPORTES E INFORMES
Necesario/Deseable	Necesario
Actor Principal:	Administrador del Sistema
Resumen:	Módulo en el cual se puede seleccionar cual es el tipo de reporte que se desea generar o imprimir.
Precondiciones:	
Postcondiciones:	
Flujo Básico:	<p>Al ingresar al módulo de reportes aparecen los informes disponibles:</p> <ol style="list-style-type: none"> 1. INSCRIPCIONES: Numero de Estudiantes inscritos por jornada Record de llamadas. 2. MATRÍCULAS: Alumnos matriculados por jornada. Matriculas financiadas. 3. SEGUIMIENTO ACADÉMICO Control de Asistencia. Formato de notas. . Rendimiento x seguimiento. 4. REPORTES >> ARCA. Alumnos por programa. ARCA. Vs nivelatorio. Documentos de Estudiantes q ya presentaron exámen. 5. HORARIOS Horario por Docente. Horario por grupo 6. DOCENTES Datos Personales. 7. REPORTE FINANCIERO Informe financiero. Reporte a Tesorería.
Flujos Alternativos:	

- **Casos de Uso**

Una vez recopilados los requerimientos se crea un conjunto de escenarios que identifican una línea de utilización para el sistema y permita, dada su facilidad para entenderse, tener una idea del funcionamiento del mismo.

Figura 5. Caso de Uso: Principal

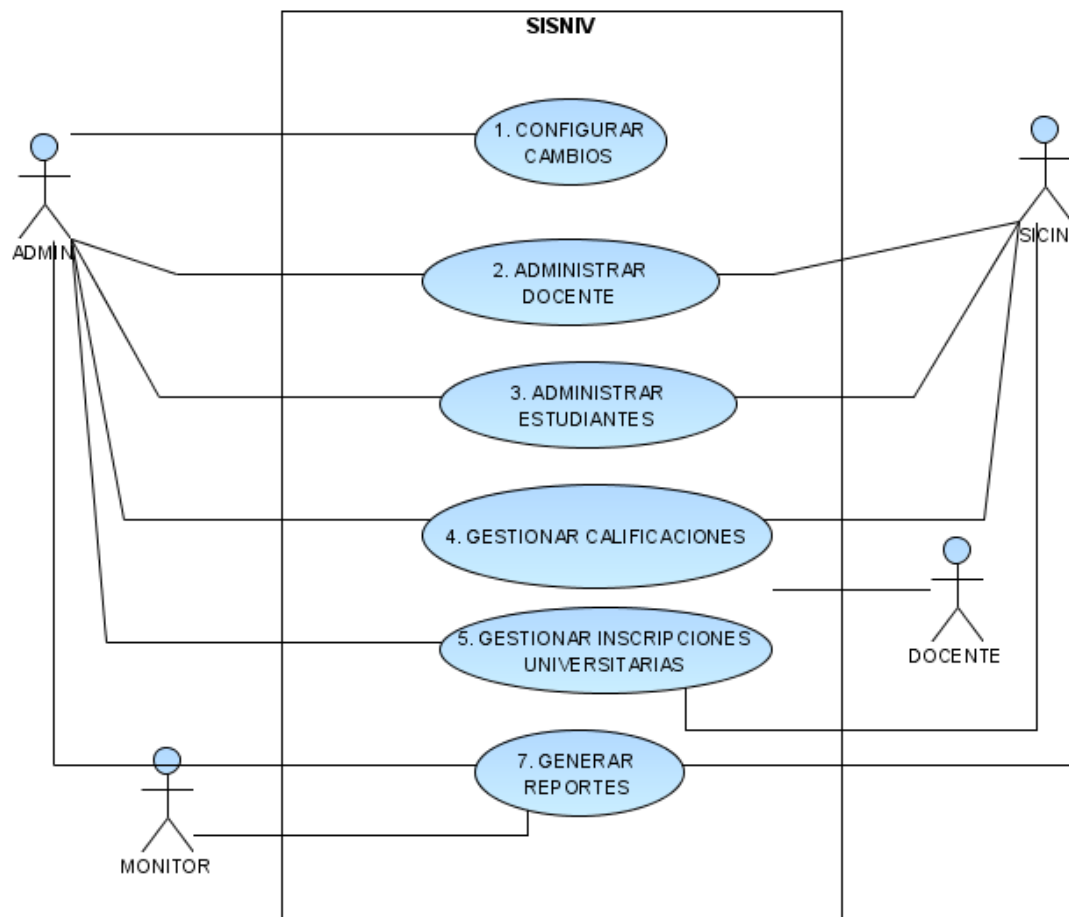


Figura 6. Caso de Uso: Configuración y Cambios

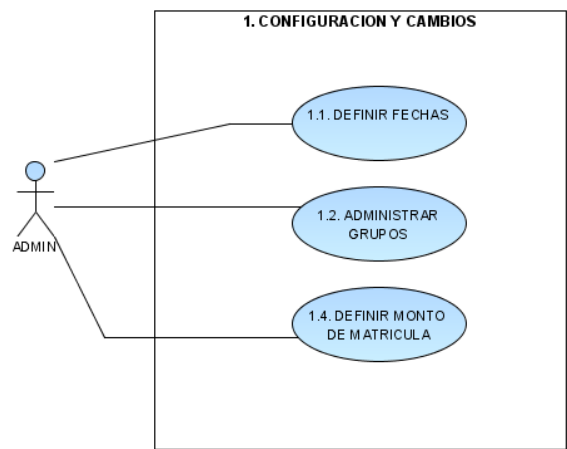


Figura 7. Caso de Uso: Administrar Docentes

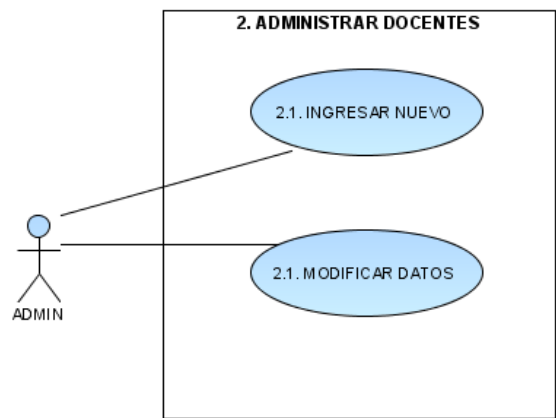


Figura 8. Caso de Uso: Administrar Estudiantes

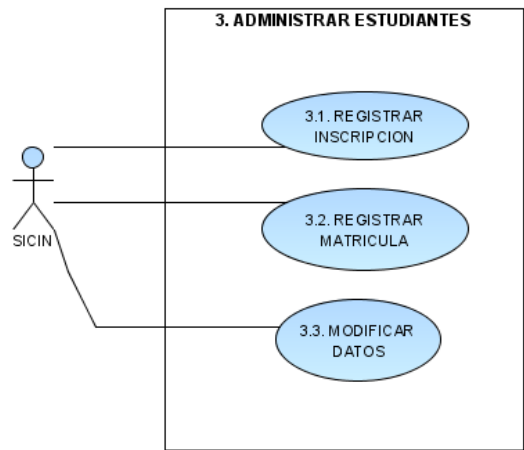


Figura 9. Caso de Uso: Gestionar Calificaciones (Docente)

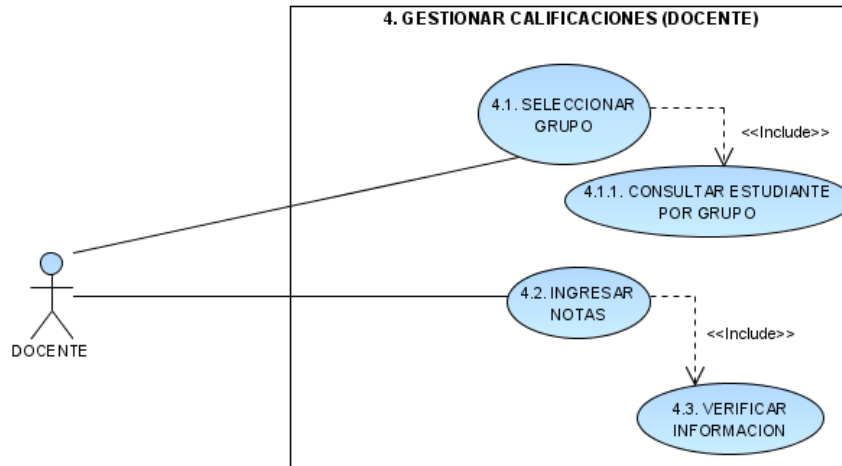


Figura 10. Caso de Uso: Gestionar Calificaciones (SICIN)

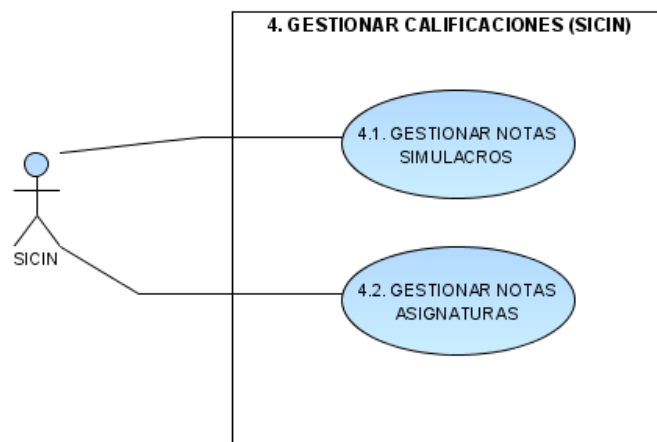


Figura 11. Caso de Uso: Gestionar Inscripciones Universitarias

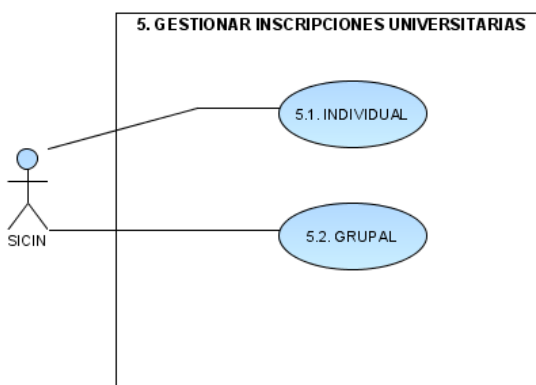
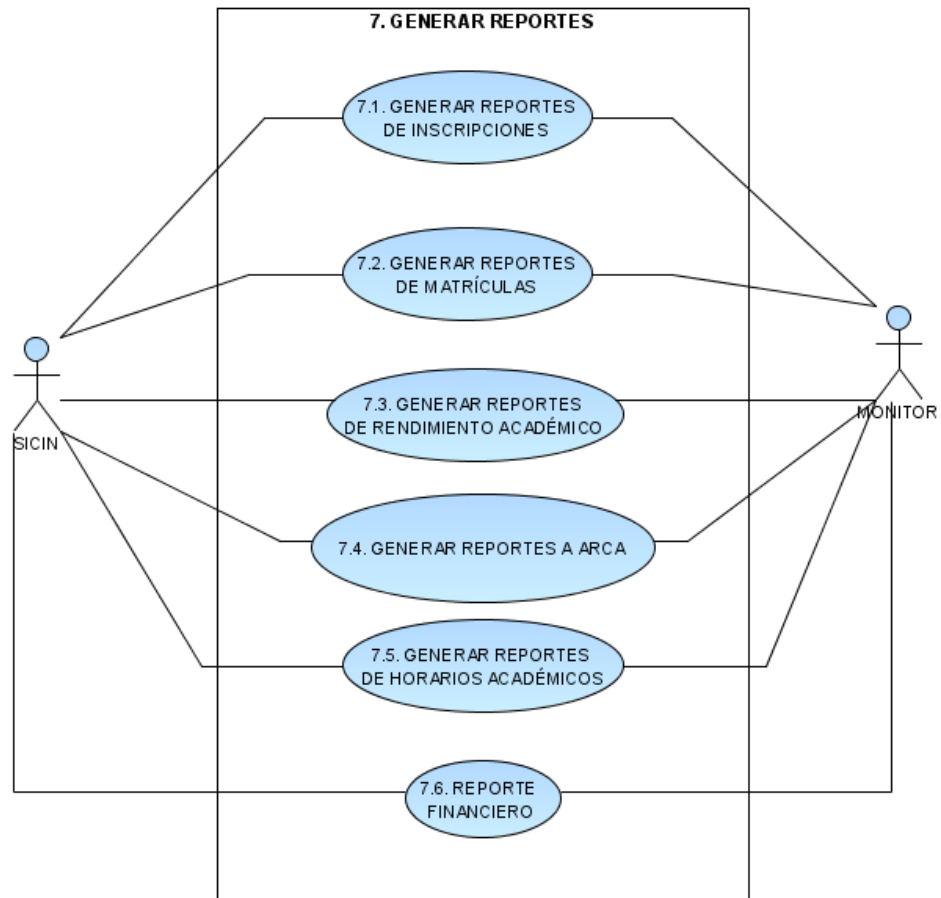


Figura 12. Caso de Uso: Generar Reportes



3.1.3. DISEÑO

En esta etapa del proyecto se utiliza la información recolectada para definir la mejor solución informática y elaborar el diseño lógico del sistema, logrando con esto proporcionar la entrada efectiva para él.

El sistema se diseñó utilizando implementando la metodología de Lenguaje Unificado de Modelamiento UML, facilitando así la elaboración de modelos del sistema que cumplan con las especificaciones y requerimientos previamente establecidos.

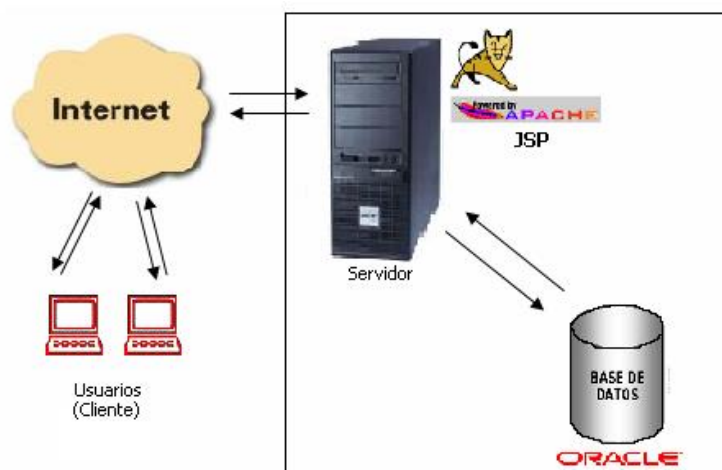
- **Arquitectura del Sistema**

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización. En otras palabras, es la estructura del sistema, lo que incluye sus componentes de software, las propiedades de dichos componentes y las relaciones entre ellos.

El sistema está desarrollado bajo la arquitectura Multicapas por todas las ventajas que éste brinda, tales como condiciones seguras para almacenamiento de datos, portabilidad, modificaciones al sistema sin que el usuario final se percate, etc.

Esta arquitectura está formada por tres capas: El servidor de aplicaciones, la base de datos y el cliente o usuario. La tecnología multicapas permite que la información esté centralizada, segura y el tiempo de respuesta sea rápido porque todos los datos están en el servidor. La arquitectura del sistema se puede apreciar mejor a través del siguiente gráfico:

Figura 13. Arquitectura web de SISNIV



- **Interfaz del Usuario**

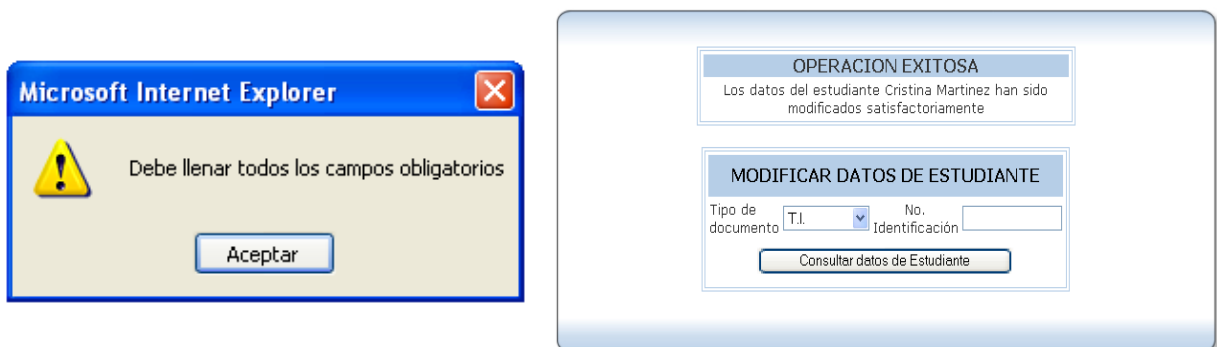
Teniendo en cuenta las especificaciones y sugerencias del usuario se realizó el diseño de la interfaz gráfica de usuario, haciendo de éste un ambiente sencillo y no recargado que contribuya a la fácil navegación a través del mismo.

Figura 14. Interfaz de usuario. Página de inicio.



En el sistema fueron diseñados mensajes de alerta, error y operación exitosa, con el fin de notificar al usuario los eventos ocurridos al operar el software.

Figura 15. Interfaz de usuario. Mensajes de alerta, error y operación exitosa.



The screenshot shows a web interface with two main sections. The top section is titled "ERROR" and contains the message: "El estudiante no se encuentra matriculado, registre la matricula antes de ingresarlo a un grupo". The bottom section is titled "INGRESAR NUEVO ESTUDIANTE A GRUPO" and contains a form with the following fields: "Tipo de documento" (a dropdown menu with "T.I." selected), "No. Identificación" (a text input field), and a "Siguiente" button.

En todos los casos, el ingreso de información al sistema, se realiza de forma fácil a través de formularios, diseñados para capturar los datos necesarios para realizar cada una de las tareas.

Figura 16. Interfaz de usuario. Inserción de datos

The screenshot shows a web interface titled "REGISTRAR NUEVA INSCRIPCION". Below the title is a section titled "Datos personales" containing the following fields: "Tipo de documento" (a dropdown menu with "T.I." selected), "No. Identificación" (a text input field), "Nombre:" (a text input field), "Apellidos:" (a text input field), "Departamento:" (a dropdown menu with "Magdalena" selected), "Ciudad" (a dropdown menu with "Santa Marta" selected), "Dirección:" (a text input field), "Telefono1:" (a text input field), "Telefono2:" (a text input field), "E-mail:" (a text input field), "Fecha de Nacimiento" (three dropdown menus for day, month, and year, with values "1", "Enero", and "1970" respectively), and "Jornada:" (a dropdown menu with "Mañana" selected). A "Registrar" button is located at the bottom right of the form.

Los reportes son presentados al usuario en pantalla con la opción de ser impresos, de tal manera de tener constancia física de los mismos.

Figura 17. Interfaz de usuario. Reportes.

[Click para imprimir](#)

REPORTE DE INSCRITOS POR JORNADA

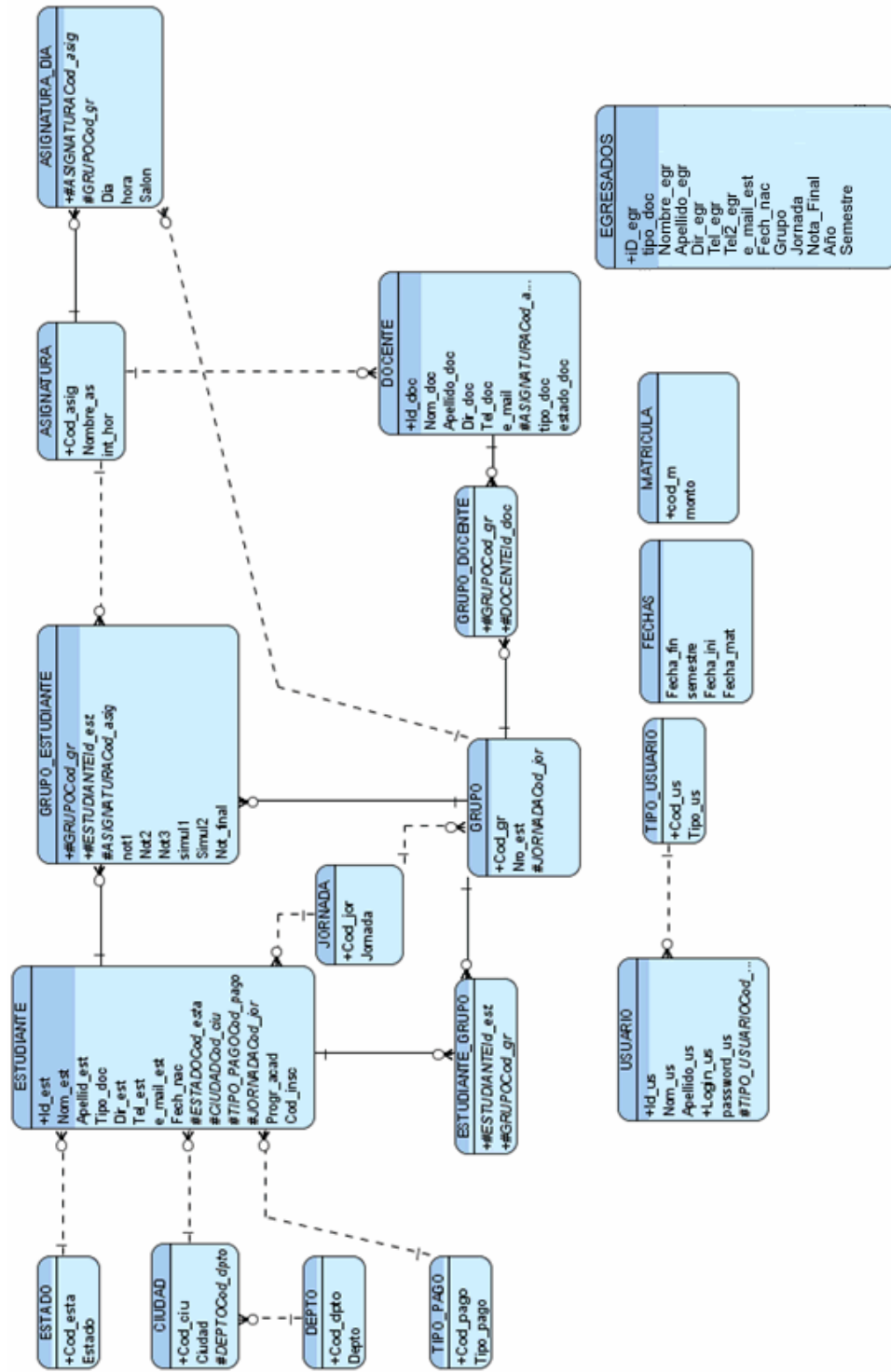
Jornada: Mañana

No.	Id. Estudiante	Nombre
1	8402206932	Ana Sofia Aguirre Ceballos
2	5498455	Josecarlo Arenilla Bustamante
3	57263102	Diana Paola Galvan Lopez
4	81151122563	Andres Mauricio Restrepo Crespo
5	82061085693	Laura Cristina Vives Pumarejo
6	85021069852	Rosa Margarita Zuñiga Ñañez

Total inscritos en la jornada Mañana: 6

Consultado por: Tatiana Restrepo Bermudez

Modelo Entidad - Relación



Diccionario de Datos

Tabla 7. Entidad Estudiante

ESTUDIANTE					
Descripción:	Esta entidad contiene la información de cada uno de los estudiantes inscritos, matriculados o egresados con sus datos personales y académicos.				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_est	VARCHAR	15	Si/principal	No	Número de identificación del estudiante
nom_est	VARCHAR	50	No	No	Primer y segundo nombre del estudiante
apellido_est	VARCHAR	50	No	No	Primer y segundo apellido del estudiante
dir_est	VARCHAR	50	No	No	Dirección de residencia del estudiante
tel_est	VARCHAR	15	No	No	Teléfono principal del estudiante
e_mail_est	VARCHAR	40	No	Si	Correo electrónico del estudiante
fecha_nac	DATE		No	No	Fecha de Nacimiento del estudiante
tipo_doc	VARCHAR	10	No	No	Identificador del tipo de Documento de Identificación del estudiante
progr_acad	VARCHAR	60	No	Si	Programa Académico al cual desea ingresar a la Universidad
cod_esta	NUMBER	3	Si/Foránea	No	Código de estado del estudiante.
cod_ciu	NUMBER	6	Si/Foránea	No	Código de la ciudad de residencia
cod_pago	NUMBER	3	Si/Foránea	No	Código de pago de la matrícula financiera
cod_jor	NUMBER	3	Si/Foránea	No	Identificador de la jornada
cod_insc	VARCHAR	15	Si/Única	Si	Número de inscripción del examen admisión anterior
tel2_doc	VARCHAR	15	No	Si	Teléfono secundario del estudiante

Tabla 8. Entidad Docente

DOCENTE					
Descripción:	Entidad que contiene los datos personales de los docentes adscritos al Ciclo Nivelatorio.				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_doc	VARCHAR	15	Si/Principal	No	Número de identificación del Docente
nombre_doc	VARCHAR	50	No	No	Primer y segundo del docente
apellido_doc	VARCHAR	50	No	No	Primer y Segundo apellido del docente
dir_doc	VARCHAR	50	No	No	Dirección de residencia del docente
tel_doc	VARCHAR	15	No	No	Número de teléfono principal del docente
email_doc	VARCHAR	40	No	Si	Correo electrónico del docente
cod_asig	NUMBER	3	Si/Foránea	No	Identificador de la asignatura que imparte
tipo_doc	VARCHAR	10	No	No	Tipo de Documento de Identificación del docente
estado_doc	NUMBER	1	No	No	Número de estado del docente: Dominio: 1 para inactivo, 2 para activo
tel2_doc	VARCHAR	15	No	Si	Teléfono secundario del docente

Tabla 9. Entidad Asignatura

ASIGNATURA					
Descripción:	En esta entidad están contenidas las asignaturas que se encuentran en el pésun académico del Ciclo Nivelatorio				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_asig	NUMBER	3	Si/principal	No	Identificador de las asignaturas
nom_asig	VARCHAR	40	No	No	Nombre de la asignatura
int_hor	NUMBER	3	No	No	Intensidad horaria semanal de la asignatura

Tabla 10. Entidad Depto

DEPTO					
Descripción:	Entidad que guarda los nombres de todos los departamentos del país				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_depto	NUMBER	7	Si/Principal	No	Número de identificación de cada departamento
departamento	VARCHAR	50	No	No	Nombre del departamento

Tabla 11. Entidad Ciudad

CIUDAD					
Descripción:	Esta entidad contiene el listado de las ciudades del país				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_ciu	NUMBER	7	Si/Principal	No	Número de identificación de la ciudad
ciudad	VARCHAR	50	No	No	Nombre de la ciudad
cod_depto	NUMBER	7	Si/Foránea	No	Identificador del departamento al que pertenece la ciudad

Tabla 12. Entidad Grupo

GRUPO					
Descripción:	Entidad que guarda la información de los grupos abiertos al público en un período académico				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_gr	NUMBER	3	Si/Principal	No	Código del grupo
nro_est	NUMBER	4	No	No	Número de estudiantes máximo en cada grupo
cod_jor	NUMBER	3	Si/Foránea	No	Identificador de la jornada a la que pertenece el grupo

Tabla 13. Entidad Jornada

JORNADA					
Descripción:	Esta entidad almacena todas las jornadas que ofrece el Ciclo Nivelatorio para realizar los cursos				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_jor	NUMBER	3	Si/Principal	No	Código identificador de la jornada. Dominio: 1 para Mañana, 2 para Intermedia, 3 para Tarde y 4 para Noche
jornada	VARCHAR	15	No	No	Nombre de la jornada

Tabla 14. Entidad Estado

ESTADO					
Descripción:	Entidad que contiene todos los posibles estados académicos de un estudiante.				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_esta	NUMBER	3	Si/Principal	No	Código identificador de los tipos de estados. Dominio: 1: Inscrito, 2: Matriculado
estado	VARCHAR	15	No	No	Nombre del estado

Tabla 15. Entidad Tipo_pago

TIPO_PAGO					
Descripción:	Esta entidad contiene los diferentes tipos de pago con los que se puede cancelar la matrícula financiera.				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_pago	NUMBER	3	Si/Principal	No	Código identificador del tipo de pago
tipo_pago	VARCHAR	30	No	No	Nombre del tipo de pago

Tabla 16. Entidad Fecha

FECHA					
Descripción:	Esta entidad contiene las fechas de realización de procesos en un período académico				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
fecha_ini	DATE		No	Si	Fecha de inicio de clases
semestre	NUMBER	2	No	No	Semestre académico
fecha_mat	DATE		No	No	Fecha límite de matrículas
fecha_fin	DATE		No	Si	Fecha de terminación de semestre

Tabla 17. Entidad Estudiante_grupo

ESTUDIANTE GRUPO					
Descripción:	Esta tabla posee la relación entre un estudiante y el grupo al que pertenece				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_est	VARCHAR	15	Si/Principal	No	Identificador del estudiante
cod_gr	NUMBER	3	Si/Principal	No	Identificador del grupo al cual pertenece

Tabla 18. Entidad Grupo_docente

GRUPO DOCENTE					
Descripción:	Esta entidad contiene la relación entre un docente y el grupo o los grupos a los cuales ha sido asignado				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_doc	VARCHAR	15	Si/Principal	No	Identificador del docente
cod_gr	NUMBER	3	Si/Foránea	No	Identificador del grupo
cod_asig	NUMBER	3	Si/Foránea	no	Identificador de la asignatura que dicta el docente

Tabla 19. Entidad Asign_Grupo_Dia_Hora

ASIGN_GRUPO_DIA_HORA					
Descripción:	En esta entidad se almacena el horario en el cual recibe clases un grupo				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_asig	NUMBER	3	No	no	Identificador de la asignatura
cod_gr	NUMBER	3	No	no	Código identificador del grupo
id_doc	VARCHAR	15	Si/Principal	no	Identificador del docente
día	VARCHAR	15	Si/Principal	no	Día asignado al grupo para esa asignatura
hora	VARCHAR	3	Si/Principal	no	Hora asignada al grupo para el día y la asignatura
salon	VARCHAR	15	No	No	Número del salón correspondiente

Tabla 20. Entidad Nota

NOTA					
Descripción:	Entidad que guarda el registro de las calificaciones de los estudiantes para cada asignatura				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_gr	NUMBER	3	No	No	Identificador del grupo
id_est	VARCHAR	15	Si/Principal	No	Identificador del estudiante
cod_asig	NUMBER	3	Si/Foránea	No	Identificador de la asignatura
nota1	NUMBER	3	No	No	Nota correspondiente al primer seguimiento
nota2	NUMBER	3	No	No	Nota correspondiente al segundo seguimiento
nota2	NUMBER	3	No	No	Nota correspondiente al segundo seguimiento
nota3	NUMBER	3	No	No	Nota correspondiente al tercer seguimiento
simul1	NUMBER	3	No	No	Nota correspondiente al primer simulacro
simul2	NUMBER	3	No	No	Nota correspondiente al primer simulacro
nota_final	number	3	No	No	Sumatoria de Notas

Tabla 21. Entidad Matrícula

MATRICULA					
Descripción:	Esta entidad contiene el costo de la matricula para el periodo académico vigente				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_m	NUMBER	2	Si/Principal	No	Identificador del monto de matrícula
monto	NUMBER	7	No	No	Valor de la matrícula financiera

Tabla 22. Entidad Habilitarpage

HABILITARPAGE					
Descripción:	En esta entidad se guarda el permiso de ingreso de la página calificaciones				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
habilitar	NUMBER	1	No	No	Bandera de habilitación de la página fechas

Tabla 23. Entidad Usuario

USUARIO					
Descripción:	Entidad que almacena los datos personales de los usuario con acceso al sistema				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_us	VARCHAR	15	Si/Principal	No	Número de identificación de un usuario
nom_us	VARCHAR	50	No	No	Nombre del usuario
apellido_us	VARCHAR	50	No	No	Apellido del usuario
login_us	VARCHAR	20	Si/Única	No	Alias de acceso al sistema
password_us	VARCHAR	30	No	No	Contraseña de acceso del usuario
cod_us	NUMBER	3	Si/Foránea	No	Identificador del tipo de usuario

Tabla 24. Entidad Tipo_Usuario

TIPO_USUARIO					
Descripción:	Esta entidad contiene los posibles roles o papeles que pueden ejercer los usuarios del sistema, tales como admin, monitor, docente, etc.				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
cod_us	NUMBER	3	Si/Principal	No	Código que identifica el tipo de usuario
tipo_us	VARCHAR	20	No	No	Nombre del rol

Tabla 25. Entidad Egresados

EGRESADOS					
Descripción:	Esta entidad almacena la información personal y académica de los egresados del Ciclo Nivelatorio				
CAMPOS					
Nombre	Tipo	Tamaño	Clave/Tipo	NOT NULL	Descripción
id_egr	VARCHAR	15	Si/principal	No	Número de identificación del egresado
tipo_doc	VARCHAR	10	No	No	Tipo de Documento de Identidad
nombre_egr	VARCHAR	50	No	No	Primer y segundo nombre del egresado
apellido_egr	VARCHAR	50	No	No	Primer y segundo apellido del egresado
dir_egr	VARCHAR	50	No	No	Dirección de residencia
tel_egr	VARCHAR	15	No	No	Teléfono principal del egresado
tel2_egr	VARCHAR	15	No	No	Teléfono secundario del egresado
e_mail_est	VARCHAR	25	No	Si	Correo electrónico
fech_nac	DATE		No	No	Fecha de Nacimiento
grupo	NUMBER	3	No	No	Número del grupo al cual perteneció
jornada	VARCHAR	15	No	No	Jornada a la cual perteneció
nota_final	NUMBER	3	No	No	Nota final que obtuvo en el curso
año	NUMBER	4	No	No	Año en el cual realizó el curso
semestre	number	2	No	No	Semestre en el cual realizó el curso.

3.1.4. CODIFICACIÓN

En esta fase del proyecto se escogen las herramientas para llevar a cabo la ejecución de los modelos descritos en el diseño, transformándolo un nivel más visible y aplicado.

El servidor de Base de Datos utilizado fue el Sistema manejador de Base de Datos ORACLE 9i, debido a su rendimiento, confiabilidad y seguridad, además de ser el manejador de base de datos implementado en los proyectos adscritos al CIDS y la Universidad posee la licencia para su utilización.

El servidor Web encargado de la aplicación de SISNIV es el servidor TOMCAT, THE APACHE JAKARTA PROJECT, que hoy en día es el servidor Web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales y permite trabajar con páginas JSP, lenguaje de implementación del sistema, además de ser totalmente gratuito, se distribuye bajo la licencia Apache Software License, que permite la modificación del código.

3.1.5. PRUEBA

Habiendo culminado con la codificación del sistema se realizaron diferentes validaciones y comprobaciones, para someter a prueba los diferentes procesos del software, con el objetivo de exponer los defectos latentes en él antes de ser entregado al usuario final.

Dentro del proceso de prueba, se enfrentó al sistema a una serie de valores inválidos con el fin de observar su reacción ante estos eventos. Seguidamente se sometió al software a información real con el fin de evaluar la funcionalidad del mismo.

Durante esta etapa, se enfrentó el sistema al usuario final, con el fin de que éste evaluara los diferentes módulos del sistema y verificara el cumplimiento de los objetivos del mismo.

4. CRONOGRAMA DE ACTIVIDADES

Fecha de inicio del proyecto : 18 de Julio de 2005
 Fecha de fin del proyecto : 6 de Febrero de 2006

Tabla 26. Cronograma de actividades

No.	Actividad	Duración	Inicio	Fin
1	Análisis y requisitos del proyecto	10 días	18/07/2005	09/08/2005
2	Reconocimiento y evaluación del proyecto	2 días	18/07/2005	20/07/2005
3	Recolección de información	8 días	21/07/2005	29/07/2005
4	Revisión de archivos y documentos	3 días	30/07/2005	02/08/2005
5	Entrevistas	5 días	03/08/2005	09/08/2005
6	Diseño del Sistema	45 días	10/08/2005	05/10/2005
7	Selección de herramientas y metodología de trabajo	2 días	10/08/2005	12/08/2005
8	Diseño del modelo de la base de Datos	5 días	13/08/2005	19/08/2005
9	Diseño del diccionario de datos	2 días	20/08/2005	22/08/2005
10	Identificación de Usuarios y Roles	1 día	23/08/2005	23/08/2005
11	Especificación de los casos de uso del sistema	5 días	24/08/2005	30/08/2005
12	Diagramación de los casos de uso	10 días	30/08/2005	9/09/2005
13	Realización de Diagrama de Clases	5 días	10/09/2005	15/09/2005
14	Realización de Diagrama de Secuencia	3 días	16/09/2005	19/09/2005
15	Diseño de Formularios	2 días	20/09/2005	21/09/2005
16	Identificación de elementos recurrentes	3 días	22/09/2005	24/09/2005
17	Diseño de navegación	2 días	25/09/2005	26/09/2005
18	Diseño de Interfaz gráfica	4 días	27/09/2005	30/09/2005
19	Diseño de menús para usuarios específicos	2 días	03/10/2005	05/10/2005
20	Desarrollo	50 días	06/10/2005	14/12/2005
21	Creación de la Base de Datos	3 días	06/10/2005	10/10/2005
22	Desarrollo del Módulo de Configuración y Cambios	18 días	11/10/2005	1/11/2005
23	Desarrollo del Módulo de Estudiantes	7 días	2/11/2005	10/11/2005

24	Desarrollo del Módulo de Docentes	11 días	11/11/2005	24/11/2005
25	Desarrollo del Módulo de Calificaciones	8 días	25/11/2005	02/12/2005
26	Desarrollo del Módulo de Reportes	10 días	03/12/2005	14/12/2005
27	Pruebas	20 días	15/12/2005	16/01/2005
28	Identificación de anomalías para cada uno de los módulos	5 días	15/12/2005	20/12/2005
29	Modificación del código	10 días	21/12/2005	10/01/2006
30	Verificación del código modificado	5 días	11/01/2006	16/01/2006
31	Documentación	105 días	11/09/2005	6/02/2006
32	Manual Técnico	18 días	12/01/2006	4/02/2006
33	Manual de Usuario	15 días	16/01/2006	3/02/2006
34	Trabajo de Grado	105 días	11/09/2005	6/02/2006

5. PRESUPUESTO

Tabla 27. Presupuesto: Total

RUBROS	TOTAL
PERSONAL	9'936.000
EQUIPOS	4'512.000
SOFTWARE	15'158.102
MATERIALES DE PAPELERÍA	111.000
TOTAL	29'717.102

Tabla 28. Presupuesto: Descripción de los gastos de personal (en miles de \$).

Nombre del Investigador / Experto/ Auxiliar	Función dentro del proyecto	DEDICACIÓN Horas	Valor Hora	TOTAL
Lorena Lara Sánchez	Desarrollador	720	5900	4'248.000
Tatiana Restrepo B.	Desarrollador	720	5900	4'248.000
Eduardo Ropain Munive	Director	96	15000	1'440.000
TOTAL				9'936.000

Tabla 29. Presupuesto: Descripción de los equipos

EQUIPO	CANTIDAD	VALOR UNITARIO	TOTAL
Computador	2	2'124.000	4'248.000
Impresora	1	264.000	264.000
TOTAL			4'512.000

Tabla 30. Presupuesto: Descripción del Software

SOFTWARE	JUSTIFICACIÓN	TOTAL
ORACLE 9i	Software manejador de Base de Datos, que brinda los niveles de seguridad requeridos para el sistema de información.	12'000.000
VISUAL PARADIGM	Software necesario para el modelado del sistema de información	3,158,102
J2EE	Paquete de herramientas, necesario para la realización de la interfaz gráfica del sistema, etc.	0
TOTAL		15'158.102

Tabla 31. Presupuesto: Materiales de Papelería

ÍTEM	CANTIDAD	VALOR UNIT	TOTAL
Papel de Impresión	4	11.500	46.000
Caja de CD RW	1	25.000	25.000
Tinta para impresora	1	40.000	40.000
TOTAL			111.000

6. CONCLUSIONES

Con la realización del Sistema de Información para el apoyo de los procesos académicos de los estudiantes del Ciclo Nivelatorio de la Universidad del Magdalena, se ofrece al personal encargado del manejo del mismo, una herramienta de software que permite realizar el registro y procesamiento de la información referente a las actividades académicas y administrativas del curso, de una manera ágil, segura y confiable.

Gracias a este sistema se pueden realizar procesos tales como, registro de inscripciones y matrículas de estudiantes de manera más eficiente y sin pérdidas de información, cumpliendo con los requerimientos establecidos por los usuarios del mismo.

Con la implementación de la página de registro de Calificaciones para docentes se logró descentralizar esta actividad, haciendo a cada docente autónomo del manejo de los grupos a su cargo, minimizando con ello la carga de labores en la coordinación del curso.

La generación de reportes a través del sistema puede ser realizada de manera rápida, obteniendo siempre información exacta, completa, pertinente, segura y oportuna para la toma de decisiones.

El uso de ORACLE como motor de base de datos del sistema de información, hace de éste un sistema seguro y confiable, asegurando el óptimo almacenamiento de los datos.

El sistema permite, a través de la documentación del mismo, el crecimiento y mantenimiento de éste, logrando que su ciclo de vida sea a largo plazo.

7. BIBLIOGRAFÍA

BOOCH, Grady y JACOBSON, Ivar. El lenguaje unificado de modelado. Addison Wesley, 1996.

BURCH, John y GRUDNITSKI, Gary. Diseño de Sistemas de Información. Teoría y Práctica. México: Grupo Noriega Editores, 1992.

DATE, C.J. Introducción a los sistemas de Bases de Datos. Volumen 1. Delaware: Addison Wesley, 1993.

ICONTEC. Tesis y otros trabajos de grado. Santa Fe de Bogotá D.C.: ICONTEC, 2002.

JOHNSON, James. Bases de datos. Modelos, lenguajes, diseño. México: Editorial Oxford, 2000.

KENDALL, Kenneth y KENDALL, Julie E. Análisis y diseño de sistemas. México: Prentice Hall, 1991.

KORTH, Henry y SILBERSCHATZ, Abraham. Fundamentos de Bases de Datos. 2da. Ed. Madrid: Mc. Graw Hill, 1994.

LARMAN, Craig. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2da. Ed. Prentice Hall, 2003.

PRESSMAN, Roger S. Ingeniería de Software. Un enfoque práctico. 5ta. Ed. Madrid: Mc. Graw Hill, 2002.

SENN, James. Análisis y diseño de sistemas de información. 2da. Ed. México: Mc. Graw Hill, 1992.

SOMMERVILLE, Ian. Ingeniería de Software. 6ta. Ed. México: Addison Wesley, 2002.

ULLMAN, Jeffrey y WIDOM, Jennifer. A first course in database system. New Jersey: Prentice Hall, 1997.

HISPALINUX. Modelado de Sistemas con UML, 2005. En: <http://es.tldp.org>

Instituto Nacional de Estadística e Informática. En: <http://www.inei.gob.pe>

ANEXOS

ANEXO A